

CREDIT CARD FRAUD DETECTION

Using Machine Learning (Random forest classifier & XGboost)

TEAM- CODEBUSTERS
THEME-
CYBERSECURITY



Problem

As technology is drastically growing, **we all are shifting to e-commerce, e-pay and so surveys say that 80-90% of the people have shifted to e-payments or online transactions**, say Paytm, Gpay or any other random surfed site of clothing or groceries. Some non-secured sites even try to access user's highly confidential information without their permission. Hackers also got their hands red into it, and can easily hack into the local sites to get the credit card informations, and we never know how **much lossy it could be!**

What Market requires?

- In today's world, fastest growing market in technical terms, it has become **a necessity** to at least have a tool or something like that, **which will let us know about our transactions.**
- A **TOOL**- which will let a user know, whether the transaction he/she made was fraudulent or not.
- Not only a particular individual needs that, in fact big firms, commercial companies, banks and many other organisation has a requirement of it.



Problem in today's industry

- There is a gap between the users and the highly secured network of transaction channels.
- There are many Hackers all around, smartly they just hack into the local sites and steal the user's credential. There are many fraud websites as well, which do the same thing.
- We all must have gone through some space of time where without any reason we get some random otps.
- And similarly, there are many different types of technique which they use, and user is unaware of that.
- Today, it is a very common practice, to steal the credentials and make money!
- So, this gap between a secured channel and the user is not yet built completely.





Magic Tool: Solution

- We are using **Machine learning techniques** and Algorithms to train a model, which will help the user to identify whether a particular transaction was fraudulent or not based on some regress patterns.
- Also, we have taken the help of **AWS (amazon web services)**, to deploy the trained model.
- The deployed model can be **used as a tool by the end users.**

Exploratory Data Analysis

- The dataset chosen is highly **unbalanced** with only 0.0172% fraudulent transactions. Hence, we analyse the data to obtain better regress patterns.

STEP 1 - Presence of any **null values** and **Skewness** of features.

STEP 2 - **Correlations** between the features and the class of the transaction (fraud or not).

STEP 3 - **Feature scaling** is done to optimise model performance.

STEP 4 - The dataset is split into X and y where y is the 'Class' (0 - non - fraud, 1 - fraud) column and X contains the rest of the dataset.

STEP 5 - X and y are further **split** into train, test and cross validation sets in suitable proportions.

Training and testing the model


STEP 1 - **Spot-Checking** technique is applied to select the best classification algorithm.

STEP 2 - Using the **cross validation** results, the algorithms which give the best accuracy are found to be Random Forest classification and XGBoost. So, we proceed to train our model with these algorithms.

STEP 3 - RF Classifier is fit to the training set and the resulting object is used to make predictions on the test set. Similarly, XGBoost Classifier is fit and predictions are made.

STEP 4 - The test set accuracies of RF and XGBoost are found. It is observed that, **XGBoost** has a slight edge over RF and hence, we use XGBoost Classifier to make our model predictions in **deployment**.

Random forest and XGboost give the highest accuracy

jupyter creditcardfraud Last Checkpoint: Last Wednesday at 16:08 (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Spot - Checking Algorithms

Spot-checking algorithms is about getting a quick assessment of a bunch of different algorithms on your machine learning problem so that you know what algorithms to focus on and what to discard.

```
In [50]: # Spot - Checking Algorithms

models = []

models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('SVM', SVC()))
models.append(('XGB', XGBClassifier()))
models.append(('RF', RandomForestClassifier()))
```

```
In [51]: # testing models

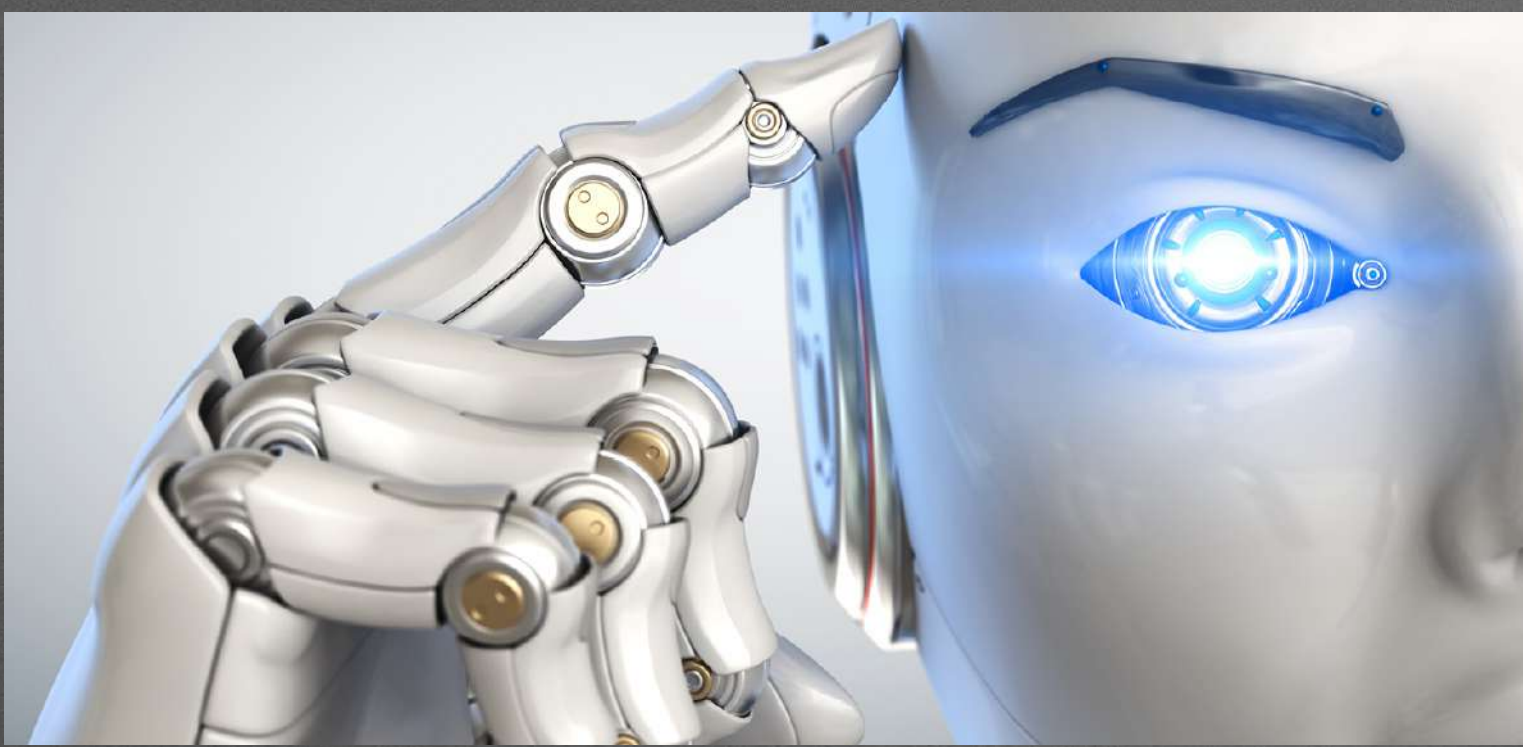
results = []
names = []

for name, model in models:
    kfold = KFold(n_splits=10, random_state=42)
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='roc_auc')
    results.append(cv_results)
    names.append(name)
    msg = '%s: %f (%f)' % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
LR: 0.979466 (0.026001)
LDA: 0.982409 (0.018455)
KNN: 0.963288 (0.041322)
CART: 0.898030 (0.067494)
SVM: 0.976313 (0.025146)
XGB: 0.978633 (0.021479)
RF: 0.974173 (0.031364)
```


Deploying the model

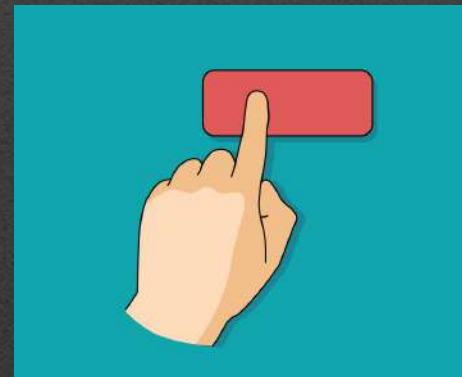
- STEP 1 -** The trained and tested model, using XGBoost, is deployed using **Flask framework** and **AWS**.
- STEP 2 -** A template html file is made to take the user inputs and give them the predicted results.
- STEP 3 -** Using the **flask framework**, we create an **API** using which we create user interface.
- STEP 4-** The inputs taken are modified and feeded to the trained XGBoost model.
- STEP 5 -** The **model predicts** whether the transaction is fraudulent or not, and accordingly informs the user (this works on a local IP address and is accessible by only us).
- STEP 6 -** Hence, to make our model **accessible to all**, the API created is **deployed using AWS**.



A step-wise briefing of the FraudDetectorTool

STEP 1 - User will have to **upload the details of the transaction**, which he/she wants to check.

STEP 2 - As soon as user clicks the PREDICT button,



- What happens in background is, it will **redirect to the trained model** that we have.
- Input entered by the user, will be converted into **desired form and fed to the ML model**.

STEP 3 - Now, on the screen USER will get the result as if the transaction was **Fraudulent or not**.

WORK FLOW



USERS/ORGANISATIONS/FIRMS

STEP 1



UPLOAD THE DETAILS
OF TRANSACTION

STEP 2

PRESS THE
PREDICT BUTTON
ON THE WEB APP

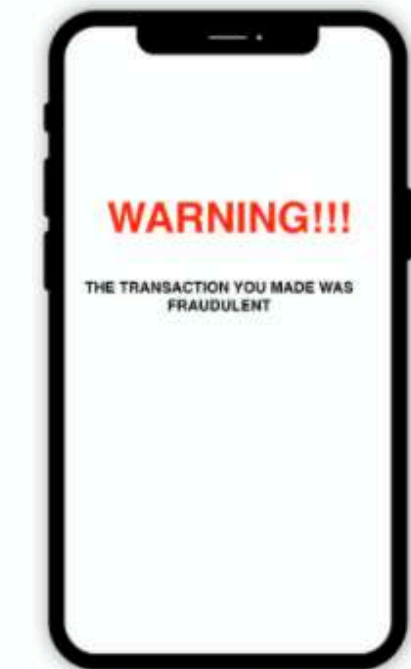
STEP 3



ML MODEL WILL CONVERT THE UPLOADED DETAILS INTO REQUIRED FORM
& PREDICT THE OUTPUT

STEP 4

OUTPUT ON THE
USER'S SCREEN





A tool for Market- Stakeholders & Firms

- The major problem nowadays is the misusing of the data, the cyber crimes that are tremendously increasing.
- So, **our tool** in the current situation would be of great use and **will fill that gap** and provide the useful.
- **Every company** from small organization to a big one, **wants to provide safe services to the customers**. Say it bank or any other e-commerce site.
- Our Tool will help **detecting the frauds** and that would be in return can **prevent a big loss to the users**

Future Scope

It is a matter of fact that, the user does not know if a fraudulent transaction is being performed using his credit card. And hence, he himself cannot give the transaction details to check its authenticity.

A **secure framework** can be built, using which the details regarding the transaction, are transferred to our **deployed model**. After encoding these confidential details using PCA, they are fed to our model to make predictions.

If the transaction is predicted to be **fraudulent**, the owner of the credit card will be **alerted**, thereby, minimising the chances of frauds.



Future Scope

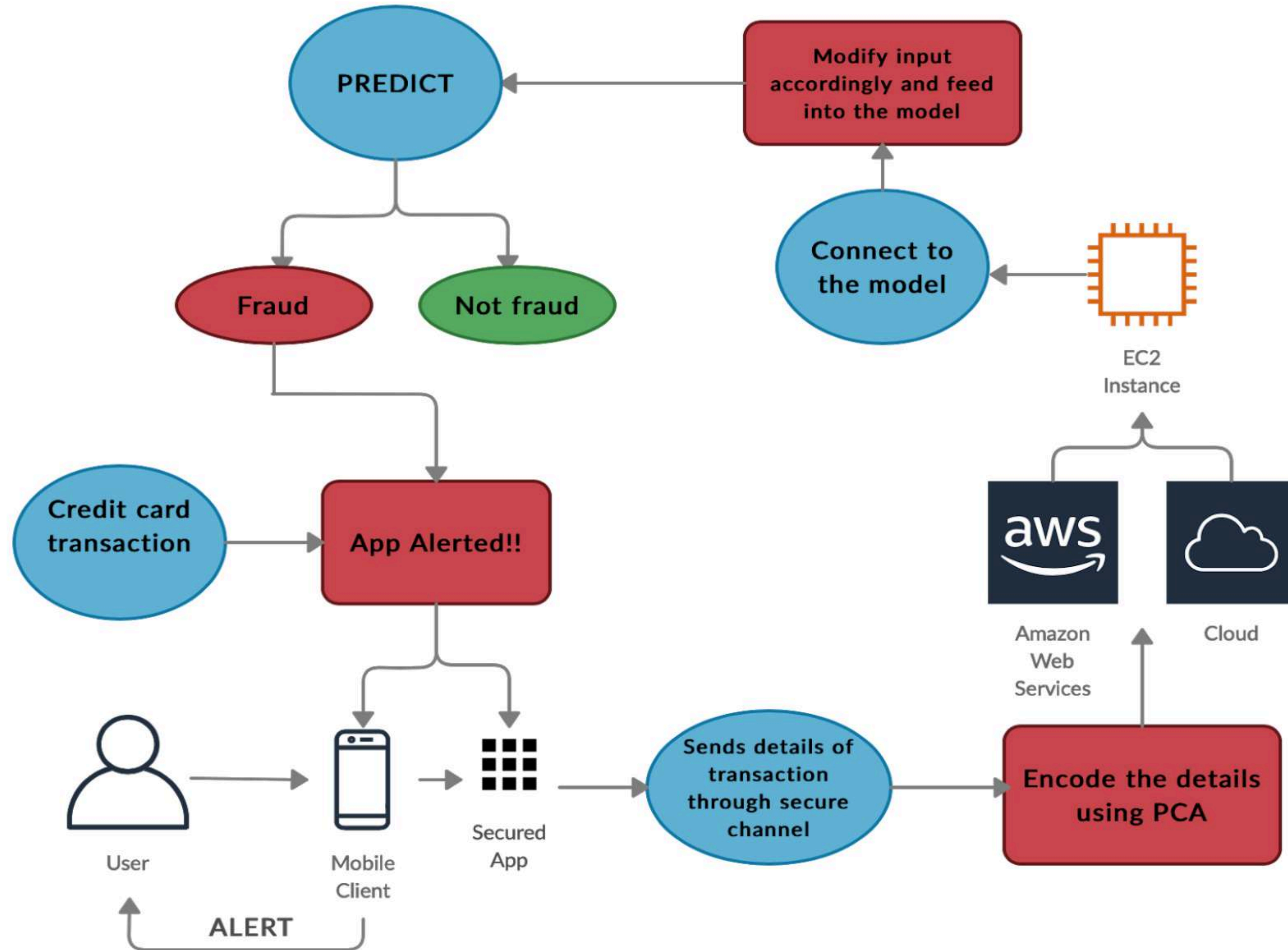
A secure app can be built for the purpose.

The credit card details of the user are integrated with the app. Whenever a transaction is done with the card, features are encoded using PCA, the app redirects to the model (cloud), checks the authenticity of the transaction and alerts the user back.

Encoding the confidential information using PCA, preserves privacy of user data.

Data privacy can also be ensured by the use of Federated learning. That is, all the user transaction data are not combined anywhere but still all the users collaboratively share our prediction model.

Future scope



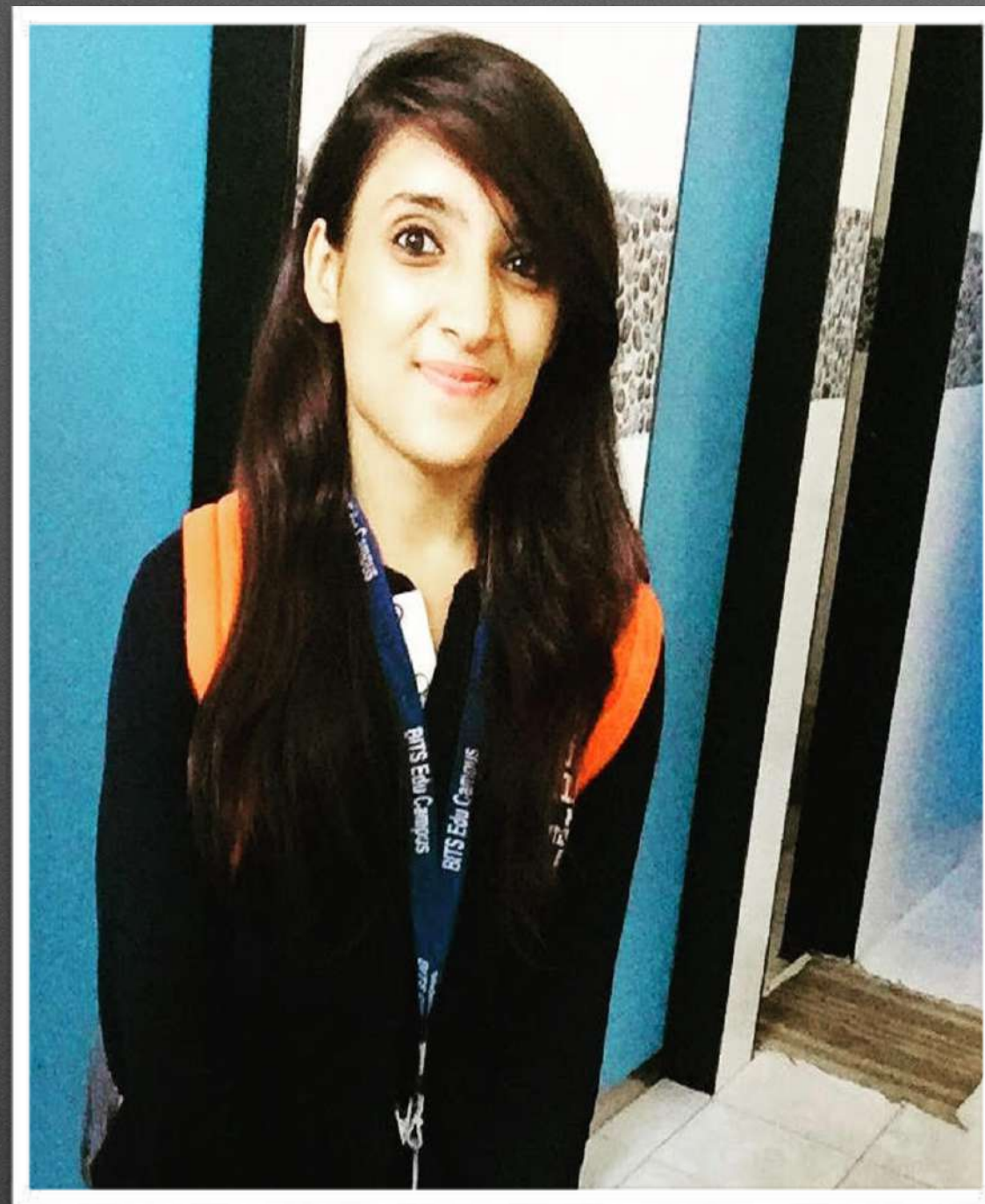
HOW LONG CAN IT SUSTAIN IN THE MARKET?

- When it comes to any Machine Learning model, it always gets better and more better as we use it time and again.
- As the tool will be used by the users, it will get more trained on the different datasets provided by them and the predictability of the model will increase.
- So better and accurate results will be predicted by the model.
- The model can learn more from the prediction results on user data and update itself accordingly (Incremental learning).

Implementation of the FraudDetectorTool

- Deployed the website for the end users (**using AWS ec2 instance**) and so it is secured at virtual amazon server.
- **[LINK TO WEB PAGE](#)**
- Jupyter notebook for ML model training-testing.
- **[GitHub LINK](#)** to the whole code and implementation including the briefed ppt-





MEDHA TIWARI
Computer science Engineering
(3rd year)

OUR TEAM



PALAKUR ESHWATHA SAI
Mathematics & Computing
(3rd year)

THANK YOU!